Proceedings of the International Conference on Aerospace Science and Technology
26-28 June 2008, Bangalore, India

INCAST 2008-009

COMPACT FIELD PROGRAMMABLE GATE ARRAY (FPGA) CONTROLLER FOR AIRCRAFT / AEROSPACE STRUCTURES

Shashikala Prakash[1], D.V. Venkatasubramanyam[1], Bharath Krishnan[1] & R. Nagendra[2]
[1]Structural Technologies Division, National Aerospace Laboratories, Bangalore, India,
shaship@css.nal.res.in, , dvvsm@css.nal.res.in, bk_13_85@yahoo.com
[2] Robert Bosch, Bangalore, India, nagendra.r@in.bosch.com

ABSTRACT: The reduction of vibration in Aircraft/Aerospace structures as well as helicopter fuselage is becoming increasingly important. The active control techniques [1] using adaptive digital filters [2] are very much suitable & well proven. Hitherto this was being achieved using high speed Digital Signal Processors (DSPs). But the throughput requirements of these systems have strained the abilities of general purpose DSPs. The Field Programmable Gate Arrays (FPGAs) have emerged as an alternative to standard DSPs, since they incorporate arrays of dedicated multipliers, embedded memory & high speed I/O. The silicon resources of an FPGA lead to staggering performance gains – while the fastest general purpose DSP can deliver up to 5 billion MAC/s (multiply-accumulate per second), leading FPGA devices can deliver more than 500 billion MAC/s – that's more than 100x faster.

In the present paper attempt is made to realize the Active Controller using the Xilinx System Generator [3] on VIRTEX - 4 FPGA. System Generator is a DSP design tool from Xilinx that enables the use of The MathWorks model-based design environment, Simulink for FPGA design. Designs are captured in the DSP friendly Simulink modeling environment using a Xilinx specific blockset. All of the downstream FPGA implementation steps including Synthesis, Place and route are automatically performed to generate an FPGA programming file. Single channel Adaptive filter & system identification has been successfully implemented & ***Hardware Co – Simulation*** has been carried out on Virtex-4 FPGA [4] using the XtremeDSP Development kit [5]. The methodology & results of this novel study are brought out in this paper.

1. INTRODUCTION

The traditional way of describing FPGA designs using hardware description languages (HDL) is too time consuming and prevents the communication between the hardware designer and the algorithm developer. The system-level design tools based on MATLAB/ Simulink, such as ***System Generator*** from Xilinx help to address this issue. The advantage is that there is no need to know HDLs. After a design is described in MATLAB/Simulink, the tools can automatically translate it into the corresponding HDL implementation. Also MATLAB/Simulink can be used to perform arithmetic-level simulation, which is much faster than the behavioral and architectural simulations in traditional FPGA design flows (ISE IDE foundation from Xilinx). These tools support hardware-in-the-loop simulation which is useful for further verification on the actual hardware.

2. WHY FPGAs?

Adaptive filters[2] which are the heart of Active Vibration Control System are traditionally implemented using DSP processors due to their ability to perform fast floating-point arithmetic. General-purpose DSPs are optimized for common Signal Processing operations. DSP systems based on software are flexible, but due to sequential nature of microprocessors, suffer from insufficient processing capability. They are the fastest method to get an algorithm running because they offer a comprehensive development environment, with tools for code analysis, debugging, and rapid prototyping. Disadvantage of DSPs is that ultimately they execute

instructions serially, setting an upper limit on the chip's throughput. Fastest DSP caters for 5 billion MACs (multiply-accumulate/sec).

On the other hand, FPGAs are Reconfigurable hardware devices where in it is possible to change the topology of electronic circuits at runtime. FPGAs incorporate arrays of dedicated multipliers, embedded memory and high-speed I/O that make them ideal for DSP applications. Also in recent times due to its growing die size as well as incorporating the embedded DSP block, the FPGA devices have become a serious contender in the signal processing field. Although it is not yet feasible to use floating-point arithmetic in modern FPGAs, it is sufficient to use fixed-point arithmetic and still achieve tap-weight convergence for adaptive filters.

In transversal adaptive filters, each tap, as well as component for updating each filter coefficient, requires a multiplier and an adder. By instantiating the required number of multipliers and adders, the performance of FPGA based filter can increase significantly compared to DSP processors. FPGAs allow great deal of parallelism to be incorporated. Also the new generation of FPGAs like Virtex-4 from Xilinx has embedded DSP slices within the device, which have dedicated circuitry to perform additions and multiplications, which are fundamental for any DSP applications

3. THE LMS ALGORITHM

Adaptive filters adjust their coefficients to minimize an error signal and can be realized as (transversal) Finite Impulse Response (FIR), (recursive) Infinite Impulse Response (IIR), lattice and transform domain filters. The most common form of adaptive filter is the transversal filter using the least mean square (LMS) algorithm. LMS uses a small step-size parameter μ, input signal, along with the difference of desired signal and filter output signal to periodically calculate the update of the filter coefficients set.

3.1. LMS Equation

Each filter coefficient adaptation uses its present coefficient value, w[n], to add to the product of the step size parameter, μ, tap input x[n] and error output e[n], to obtain an updated version of the filter coefficient w[n+1]. All updated filter coefficients are then gathered to perform convolution with the taps to produce a filter output. The filter output y[n] is subtracted from the desired value d[n] to produce an error term, e[n], which is fed back into the filter coefficient update equation to produce consequent coefficient updates.

The equations are shown below:

$$e[n] = d[n] - y[n] \quad (1)$$

$$w[n+1] = w[n] + \mu * x[n] * e[n] \quad (2)$$

The only information needed to update filter coefficients are the tap input, the error term and the step-size parameter. The choice of the step-size parameter and the order of the filter effectively determine the ***performance of LMS algorithm***. Unfortunately, there is no clear mathematical analysis to derive the quantities. However the step size parameter μ is bounded in the range of $0<\mu<(2/\lambda_{max})$, where λ_{max} is the maximum eigenvalue of the auto-correlation matrix of the filter input. Also μ is a decimal number between 0 and 1.

3.2. System identification & control

An adaptive filter can be used in modeling. That is imitating the behavior of a physical dynamic system which may be regarded as unknown "black boxes" having one or more inputs and one or more outputs. In modeling a single-input, single-output dynamic system or "plant", both the unknown system and adaptive filter are driven by the same input. The adaptive filter adjusts itself with the goal of causing its output to match that of the unknown system. Once identification is complete, a modified form of LMS called Filtered – X LMS[1] is used to achieve control.

4. XILINX SYSTEM GENERATOR

System Generator [3] is a system level modeling tool that facilitates FPGA hardware design and provides high-level abstractions that are automatically compiled into an FPGA at the push of a button. The tool also provides access to underlying FPGA resources through lower level abstractions, allowing the user to implement highly efficient FPGA designs.

Programming an FPGA using System Generator means describing a computation as a Simulink model, generating a hardware description from this model and then compiling this hardware description into an FPGA configuration file, called a bitstream. The final step of compiling a hardware description into a bitstream is unique to System Generator. System Generator blocksets allow you to construct bit-accurate and cycle-accurate models of an FPGA circuit in Simulink. System Generator provides hardware co - simulation interfaces that make it possible to incorporate an FPGA directly into a Simulink simulation. The code generator has "Hardware Co - simulation" compilation targets (analogous to the HDL Netlist target) that automatically create a bitstream. After creating the bitstream, System Generator automatically incorporates an FPGA hardware platform configured with this bitstream back into Simulink as a run-time block. When the design is simulated in Simulink, results for the compiled portion are calculated in hardware. This allows the compiled portion to be tested in actual hardware and can speed up simulation dramatically.

5. XILINX VIRTEX-4 DEVICES

Virtex-4 FPGAs [4] from Xilinx have several features which are not available in earlier FPGAs. That is, they provide 2x more density and boost performance as much as 2x, while reducing power consumption by as much as 50% compared with previous-generation FPGAs. At the same time, Virtex-4 FPGAs cut the cost of programmable system platforms by more than 50%, enabling developers to adopt high-performance FPGAs in an extraordinary range of products.

Within the Virtex-4 FPGA, Xilinx has crafted the versatile XtremeDSP slice, providing twice the DSP performance of previous implementations while drawing less than 1/7th of the power. Although all Virtex-4 FPGAs contain XtremeDSP slices, the Virtex-4 SX platform provides the highest ratio of XtremeDSP slices to other resources. The largest SX device, the XC4VSX55, has 512 slices. Using these 500 MHz XtremeDSP slices with 18 x 18-bit multiplier and 48-bit accumulator exclusively, this device can achieve 256 GMAC/s performance providing the most powerful DSP capabilities of any FPGA in the industry. The DSP-optimized SX55 offers ten times the DSP value as compared with previous generation FPGAs.

In the present study The XtremeDSP Development Kit from Nallatech [5], having the SX35 Virtex-4 FPGA serves as an ideal development platform during prototyping because of the availability of dual channel high performance ADCs and DACs.

6. SIMULATION STUDIES USING SYSTEM GENERATOR

The adaptive filter (Fig.1) is comprised of the System Generator blocks delay, constant, multiplier and adder-subtractor. It is embedded in a test circuit that consists of a data source built from Simulink blocks (the sum of a high frequency and low frequency sine wave) and an output sink (scope). System Generator "gateway" blocks convert between Simulink's double-precision floating point signals and System Generator's fixed-point arithmetic type. The System Generator block provides control of system and simulation parameters and is used to invoke the code generator.

The Xilinx JTAG Co-Simulation block allows the user to perform hardware co-simulation using JTAG and a Parallel **Cable IV** or Platform USB. The custom JTAG co-simulation block with ports that match the gateway names (or port names if the subsystem is not the top level) from the original model (Fig. 2), interacts with the FPGA hardware platform during a Simulink simulation. Simulation data that is written to the input ports of the block are passed to the hardware by the block. Conversely, when data is read from the co-

simulation block's output ports, the block reads the appropriate values from the hardware and drives them on the output ports so they can be interpreted in Simulink. The simulation results as well as hardware co-simulation results are shown in Fig. 3. In Fig. 4 the system identification block along with hardware co – simulation block are shown. Fig. 5 shows the results from real time system identification.

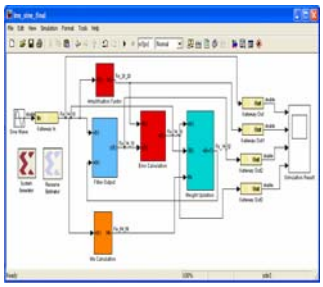

Fig.1 LMS Adaptive Filter

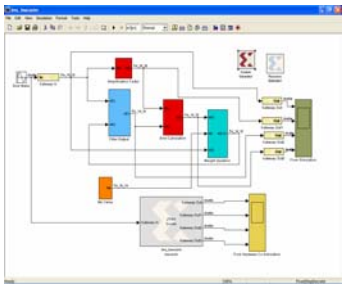

Fig. 2 LMS Adaptive Filter Hardware Co – Simulation on V4

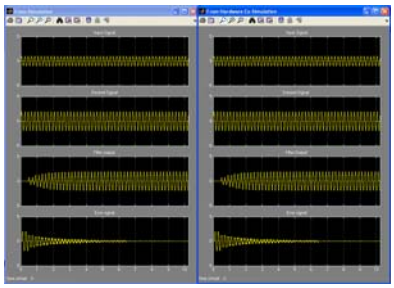

Fig. 3 Comparison of results from Simulation & FPGA

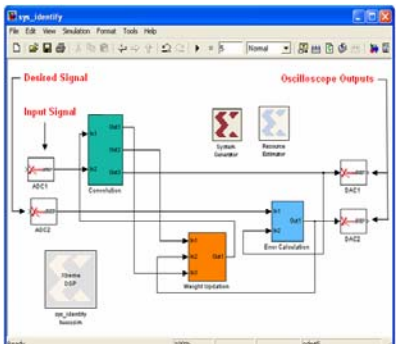

Fig.4. Real Time System Identification

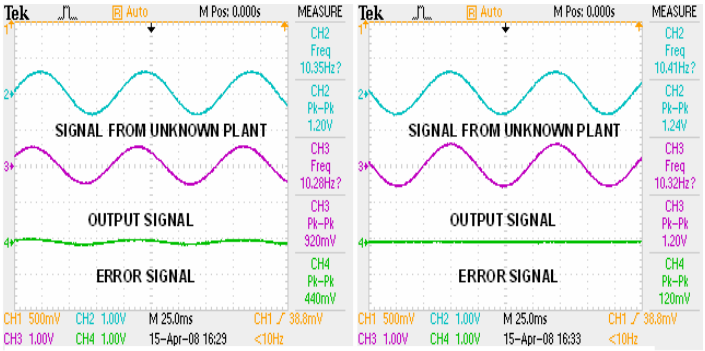

Fig. 4. Results from System identificatin Identification

CONCLUSION

The implementation of LMS adaptive filter on Virtex-4 FPGA has been presented. Hardware Co-Simulation & real time system identification has been tested successfully on the XtremeDSP Development kit.

ACKNOWLEDGMENTS:

This work is supported by the Aeronautical Research and Development Board. Our sincere thanks to The Coordinator & the panel members of AR&DB, Structures panel for giving us the project & financial support to carry out the work at NAL. The authors would like to thank Dr. A. R. Upadhya, Director, NAL for giving us all the encouragement & support in this development. The authors would like to thank Mr. Aravind V. Pavate for the technical support he extended for this work.

REFERENCES

[1]. Sen M. Kuo, Dennis R. Morgan – " ***Active Noise Control Systems – Algorithms & DSP Implementations***" , John Wiley & Sons, Inc 1996.

[2]. John R. Treichler, C. Richard Johnson, Michael G. Larimore – "***Theory and design of adaptive filters***", John Wiley& sons 1987.

[3]. Xilinx System Generator for DSP *User's Guide* November 28, 2006

[4]. **Virtex-4** : Breakthrough performance at the lowest cost, by Greg Lara, Virtex solutions, Xilinx Inc.

[5]. XtremeDSP Development Kit-IV User Guide